CHAPTER 4

PHASE I-PREPROCESSING AND INSIDER DETECTION (P&ID)

CHAPTER 4

PHASE I-PREPROCESSING AND INSIDER DETECTION (P&ID)

4.1 INTRODUCTION

The methodology of phase I - Preprocessing and Insider Detection (P&ID) is discussed in this chapter. Phase I consist of two layers namely Preprocessing, and Insider Detection. Phase I utilize publicly accessible CERT insider dataset for insider detection. The CERT insider dataset gathered from various sources and are in different format and contain imbalance data should be free from class imbalance problem. The preprocessing is done to combine the data gathered from various sources, convert the data into homogenous format and to solve the class imbalance problem. Insider Detection layer, detects and classifies the user into genuine, intentional and unintentional insiders.

The following are the major contributions of the P&ID phase:

- Explored three oversampling and five undersampling approaches, and tuned the best performing sampling approach to significantly address the class imbalance issue in the insider threat dataset.

- A hybrid B-SVM algorithm has been proposed to minimize the misclassification rate for detecting genuine, intentional and unintentional insider.

The following section discusses the methodology of P&ID phase elaborately along with the results obtained.

4.2. P&ID METHODOLOGY OVERVIEW

The following Figure 4.1 depicts the overall methodology of the P&ID phase. The P&ID phase consists of two layers, namely Preprocessing and Insider Detection. Data integration, encoding, and sampling are done in Preprocessing to process diverse categorical data and to handle the class imbalance problem. In Insider Detection, a hybrid machine learning algorithm B-SVM is proposed to detect genuine, intentional and unintentional insiders.

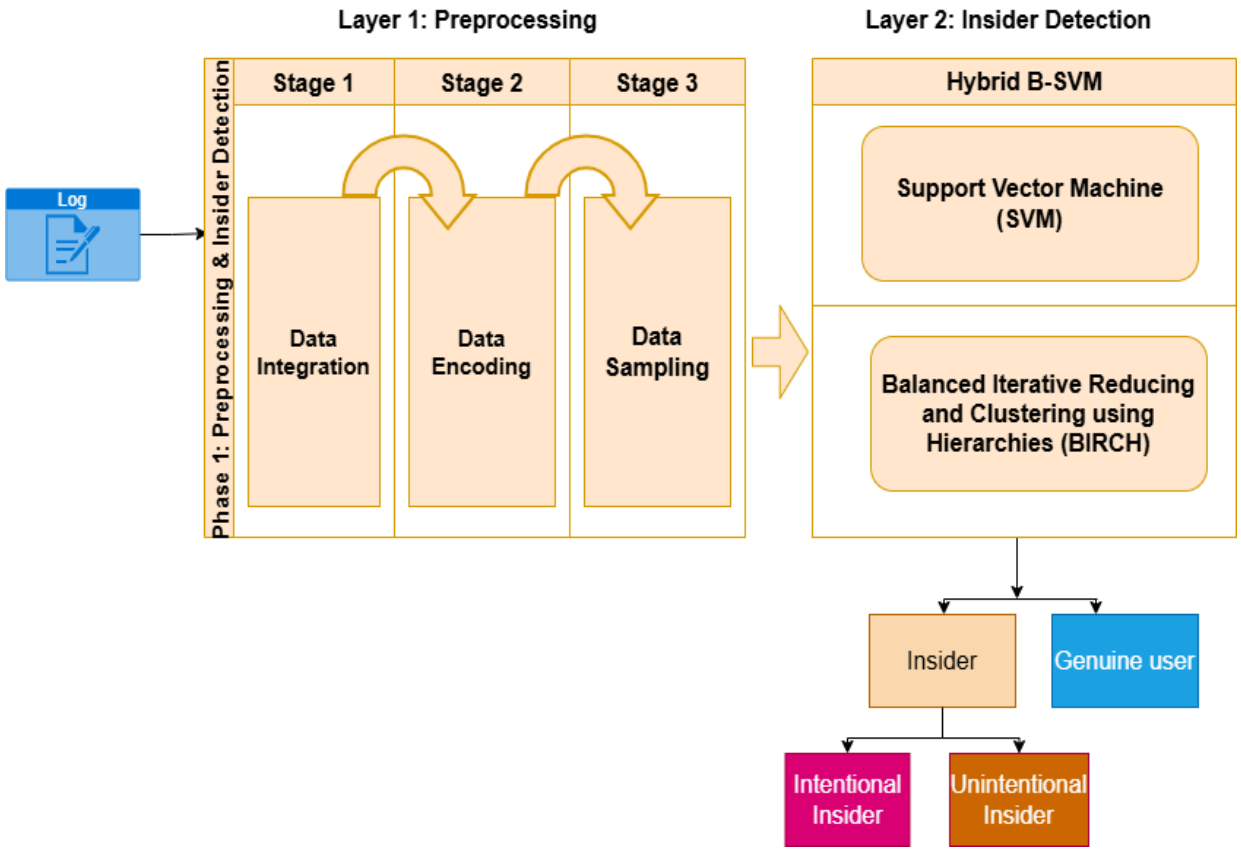

Figure 4.1: Methodology Overview of P&ID Phase

4.2.1 DATASET

The proposed methodology is evaluated and validated using two datasets.

1. The proposed methodology is evaluated using CERT insider dataset (Lindauer, 2020) (Nicolaou et al., 2020).

2. The proposed methodology is validated using CIC darknet dataset (Rust-Nguyen et al., 2023).

A. CERT Insider Dataset

Log data is necessary to analyze the user activities to detect and mitigate both intentional and unintentional insiders. The artificial synthetic insider threat data is generated in Carnegie Mellon University (CMU) by Computer Emergency Response Team (CERT) division in collaboration with ExactData, LLC (Lindauer, 2020). The dataset adopts five

sequences of abnormal events containing 250,078 log behaviors generated by 4000 users in 516 days where 30 users perform 280 malicious behaviors (Nicolaou et al., 2020). The CERT dataset contains the log events generated from signing in, website perusing, document acquiring, email addressing, analyzing psychographs, and lightweight directory access protocol (LDAP). According to CERT, the log events of insider should satisfy the five different insider threat scenarios. The five scenarios and their corresponding dataset version is detailed in table 4.1.

Table 4.1. Dataset Versions Associated with Five Scenarios.

Scenarios	Description	Corresponding Dataset
SC-1	Utilizing a USB drive after a working interval to acquire and manage sensitive documents in prohibited websites aiming for resignation.	R3.2, R5.1, R6.1, R6.2
SC-2	Obtaining sensitive data from USB and visiting employment offers in business opponent websites for increasing recruitment possibilities in a condition of sensitive credentials	R3.2, R5.1, R6.1, R6.2
SC-3	An aggrieved system administrator installs a keylogger in the supervisor's device to achieve data trafficking, sends bulk distress emails, and vacates his position.	R5.1, R6.1, R6.2
SC-4	Accessing unassigned devices by employees for regularly gathering sensitive information about others for over 3 months.	R5.1, R6.2
SC-5	Due to layoff, employees upload sensitive documents to Dropbox, planning to use them for peculiar concerns.	R6.1, R6.2

Table 4.1 shows that R3.2 version satisfies 2 scenarios, R5.1 and R6.1 satisfies 4 scenarios, and R6.2 satisfies every five scenarios. From the four version, R3.2 version is considered as standard dataset for analyzing the P&ID phase.

Attribute Overview

The dataset comprises of five attributes in R3.2 version related to log events that provide the detailed insights into characteristics of particular activity performed. These parameters include ID, Date, User id, PC id, and activity.

These attribute offers a simple-dimensional view of user behavior, capturing activity of genuine and malicious. Such details are essential to differentiate minor variations in user activity that may indicate insider threats. The dataset description of the CERT insider dataset is enumerated in table 4.2.

Table 4.2: Dataset Description of CERT Insider Dataset

Attribute	**Definition**
Id	It denotes the unique identifier for each log
Date	It denotes the date for every single log behavior
User id	It denotes the unique identification number for each user who accomplishes specific behavior
Pc id	It denotes the unique identification number for each computer where particular log behavior is performed
Activity	It denotes specific log behavior accomplished by particular user in particular computer

The log events containing genuine and malicious activities are utilized for evaluating the proposed research for detecting genuine, intentional and unintentional insiders.

B. CIC Darknet dataset:

The CIC Darknet dataset (Rust-Nguyen et al., 2023) is used for evaluating the proposed methodology. It amalgamates both ISCXTor2016 and ISCXVPN2016 datasets from the University of New Brunswick. It contains the real-time traffic information gathered using Wireshark and TCPdump through CICFlowMeter. In each session, the traffic packet contains traffic information is used to extract the traffic samples. In total, the dataset involves 158,659 labeled samples. The label contains top level traffic information and categorized into Tor, non-Tor, VPN, and non-VPN.

The CIC darknet dataset is used to analyze and detect genuine, intentional and unintentional insider activities in anonymized networks. Because, it captures a wide range of network behaviors across diverse traffic categories present in a concealed network settings.

Attribute Overview

The dataset comprises log activities with 85 distinct parameters that provide detailed perceptions of network flows, packet characteristics, and protocol-specific behaviors. It also contains source and destination IP addresses, ports, protocol type, and dates. Other parameters includes packet length, header information, and packet transmission rates.

These features offer a high-dimensional view of network behavior that captures flow pattern and individual packet level. Such granular details in user activity is required to distinguish the genuine activity and insider threats. The dataset description of the CIC darknet dataset is listed in table 4.3.

Table 4.3. Dataset Description of the CIC Darknet Dataset

Parameters	Attribute(s)	Definition
Flow Parameters	'Flow ID', 'Flow Duration', 'Flow Bytes/s' and 'Flow Packets/s'	It describes the overall data transfer patterns to distinguish abnormal flows associated with potential insider threats.
Forward and Backward Packet Metrics	'Total Fwd Packet', 'Total Bwd Packets', 'Total Length of Fwd Packet', and 'Total Length of Bwd Packet'	It defines packet characteristics in bi-direction, which helps identify anomalies based on variations in packet flow.
Inter Arrival Time Metrics	'Flow IAT Mean', 'Flow IAT Std', 'Flow IAT Max', and 'Flow IAT Min'	It is used to recognize the unusual delays or bursts in network traffic.
Flag Counts and Ratios	'FIN Flag Count', 'SYN Flag Count', 'RST Flag	It gives signal specify packet functions within TCP protocols and help identify

Parameters	Attribute(s)	Definition
	Count', 'PSH Flag Count', 'ACK Flag Count', and 'URG Flag Count'	unusual or unauthorized network activities.
Additional Traffic Indicators	'Down/Up Ratio', 'Average Packet Size', 'Fwd Segment Size Avg', and 'Bwd Segment Size Avg'	It supports a deeper analysis of normal versus abnormal traffic patterns.
Subflow Metrics	'Subflow Fwd Packets', 'Subflow Fwd Bytes', 'Subflow Bwd Packets', and 'Subflow Bwd Bytes'	It represents the smaller flows within a larger data flow
Active and Idle Time Metrics	'Active Mean', 'Active Std', 'Active Max', 'Active Min', 'Idle Mean', 'Idle Std', 'Idle Max', and 'Idle Min'	It represents the active user activity and inactivity periods.
Labels and Classification	Non-Tor, NonVPN, VPN, or Tor	It provides label of different types of traffic

The samples in top level traffic category labels are further subcategorized in terms of application types utilized to create network traffic at the application level. The subcategories includes VOIP, video streaming, P2P, file transfer, email, chat, browsing and audio streaming. The application classes of network traffic in CIC darknet dataset is detailed in Table 4.4.

Table 4.4: CIC Darknet Application Classes (Rust-Nguyen et al., 2023)

Application class	**Applications considered**
VOIP	Hangouts, Facebook and Skype
video streaming	YouTube and Vimeo
P2P	BitTorrent and uTorrent
file transfer	Skype and FileZilla
Email	SMTPS, POP3, and IMAPS
Chat	ICQ, AIM, Skype, Facebook, and Hangouts
browsing	Chrome and Firefox
audio streaming	YouTube and Vimeo

Correspondingly, the information regarding users' digital behavior in the darknet is utilized in this research phase to originate intentional and unintentional insider threats implementing diverse traffic, including The Onion Router and Virtual Private Network, as mentioned in Table 4.4.

4.2.2 PREPROCESSING AND INSIDER DETECTION (P&ID)

Log data containing the cyber activities of users from diverse sources are first processed using three preprocessing techniques and the outcome of preprocessing is a structured and well-balanced data and free from the class imbalance problem. Then, the balanced data is trained and classified using the B-SVM technique in Insider Detection for detecting genuine, intentional, and unintentional insiders.

The proposed methodology for phase I comprises of two layers. In the first layer, Preprocessing is encompassed using integration, encoding, and sampling techniques to preprocess and handle the class imbalance problem. In the second layer, the novel hybrid algorithm namely B-SVM is proposed to detect and classify users into genuine, intentional, and unintentional insiders based on abnormal behavior patterns. The working procedure involved in the P&ID phase is tabulated in Table 4.5.

Table 4.5: Working Procedure of P&ID Phase

Layer 1- Preprocessing
Step 1: Combine the log data captured from all sources (Data integration) Step 2: Transform the features in categorical format into numerical values (Data encoding) Step 3: The majority class instance with respect to minority class instance is balanced to combat imbalanced data (Data sampling)
Layer 2- Insider Detection
Step 4: The sampled data is trained using proposed B-SVM algorithm to detect genuine, intentional, and unintentional insiders. i.e., Initially, the SVM algorithm is used for intentional insider detection that recognizes genuine and intentional insiders based on user activity. The BIRCH algorithm is used for unintentional insider detection that analyses falsely detected intentional insider activities from SVM algorithm and classifies them into genuine, intentional, and unintentional insiders (Insider Detection)

Layer 1: Preprocessing

The log behavior gathered from various sources requires Preprocessing to obtain the homogenous numerical log activities containing well-sampled class instances. Figure 4.2 illustrates the overview of Preprocessing layer.

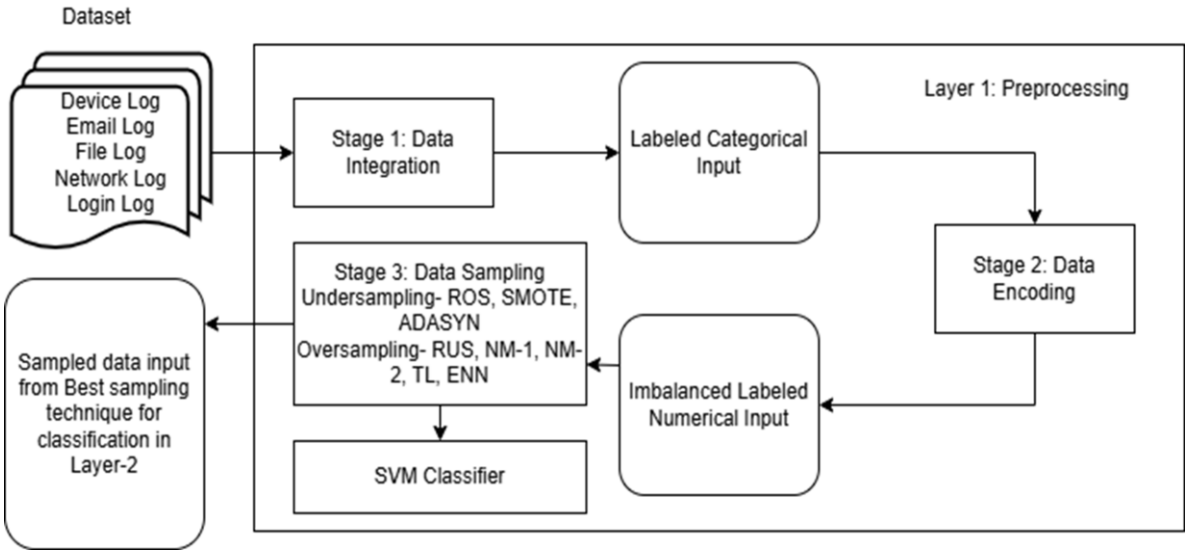

Figure 4.2. Overview of Preprocessing Layer

The log information from five various sources are combined in Data integration. The outcome of Data integration is integrated data containing categorical value. In Data encoding, the combined categorical data is encoded to obtain the imbalanced labelled numerical data. The imbalanced data undergo Data sampling to handle the class imbalance problem. Diverse sampling approaches are trained with SVM classifier to find the higher performing sampling approach. The sampled data is used to analyze the insider threat pattern in layer 2.

The following subsection discusses the three stages of preprocessing techniques: Data integration which combines diverse source data, Data encoding that converts the categorical value, and the sampling approaches which handles the class imbalance problem in insider threat data.

i) Data integration

The CERT dataset contains required log information of individual working behavior gathered from various sources, including network enrolling (Logon), extrinsic peripheral device attachment (Device), surfing external websites (HTTP), accessing documents (File), and sharing content (Email). The CERT dataset utilized in this phase that satisfies five scenarios are combined and recognized as a benchmark dataset. Table 4.6 provides a detailed description of log files from various sources and the sample dataset is enclosed in Annexure I.

Since the log information is gathered from five different sources, as mentioned above, the data remains unstructured, which reduces data quality standards, resulting in poor performance for detecting insider threats. Therefore, this information must be homogenized into similar isolated records by incorporating a simple feature concatenation algorithm in Data integration (Padmavathi et al., 2022) (Zhu et al., 2020).

Figure 4.3 illustrates the process of Data integration from multiple sources. Data is integrated based on common attributes namely date, user id, pc id, and activity to create a baseline unique table. Meanwhile, the description is inapplicable due to the disserted value. However, two features namely insider threat and source are considered as a part of baseline data. Because, feature specifying genuine user and insider is considered as Insider Threat. The feature ‘source’ denotes the source of log information is useful for further processing

Table 4.6. Detailed Description for CERT Log Files.

File/Source	Description
Logon.csv	A file that records the employee activity, such as logon/logoff in an organization. It includes the user id, pc id, and log events with the respective dates.
File.csv	A file that records file operation concerning read, update, and transfer of data to removable devices. It includes user id, pc ids, file type, content, and date.
HTTP.csv	A file that records logs related to the web browsing of employees. It includes user id, pc id, web activity, visited URL, and date.
Email.csv	A file that records the logs with respect to the mailing service. It includes user ID, pc id, email address, email size, attachment applicable, and date.
Device.csv	A file that records log information related to removable device usage, whether connected or disconnected. It contains user id, pc id, device id, device event, and date.

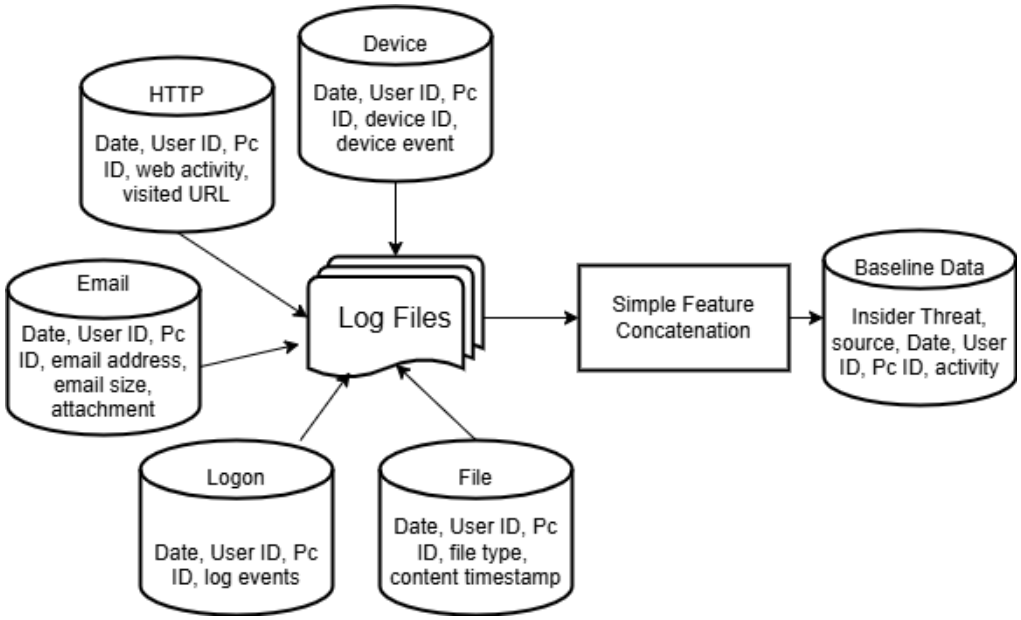

Figure 4.3: Data Integration from Multiple Sources

Figure 4.3 shows that the integrated data contains the attributes of Insider Threat, source, Date, User id, Pc id, and Activity in a structured homogenous format.

ii) Data encoding

After Data integration, Data encoding is performed. Sine, the integrated data is obtained by combining log information from various sources contains categorical variables, it requires Data encoding. Table 4.7 depicts the feature description of integrated data before Data encoding.

Table 4.7. Feature Description of Integrated Data Before Data Encoding.

Feature	**Description**
user id	Categorical
pc id	Categorical
date	Ordinal
activity	Categorical
insider threat	Categorical

From Table 4.7, it is evident that every feature in the combined baseline data is categorical, excluding the date, which is ordinal and contains a numerical value. Hence, Data encoding is required to ensure data quality standards for optimizing the ML performance. If Data encoding is not performed, the ML model will fail to recognize the relationship between data points and misinterpret results for insider threat detection. Thus, Data encoding becomes mandatory for processing feature variables to combat such challenges.

Since the baseline dataset contains a datatype of categorical and ordinal, multiple categorical encoding techniques can be used to address categorical variables. In this stage, the label encoding technique is used to assign a numerical value for every categorical feature. Meanwhile, the ordinal data feature, namely ‘date’, is encoded by shaping them into Unix Epoch Time format. This results in non-complex computation and can store dated information than conventional date systems (Al-Shehari & Alsowail, 2021). Table 4.8 displays the feature description of integrated data after Data encoding.

Table 4.8. Feature Description of Integrated Data After Data Encoding.

Encoded Feature	Description
user id	Integer
pc ID	Integer
date	Integer
activity	Integer
insider threat	Integer

iii) Data sampling

After Data encoding, Data sampling is performed to handle the class imbalance problem in encoded data. i.e., the 'insider threat' feature contains more number of instance in genuine activity and less number of instances in insider activity. A class imbalance problem is an unequal data distribution that arise when the total number of instances within one class significantly lesser than another. As a result, the performance of classification by the machine learning model would be misinterpreted. In this research, data level sampling approaches (Pengfei et al., 2014) (Maciejewski & Stefanowski, 2011) (Zhang & Zhang, 2016) (Lin et al., 2014) (Padmavathi et al., 2020b) are applied to solve the challenges of class imbalance problem. In data level sampling, the sampling approaches are classified into oversampling and undersampling (Zhu et al., 2020) (Wang et al., 2020). Figure 4.4 depicts the classification of sampling approaches.

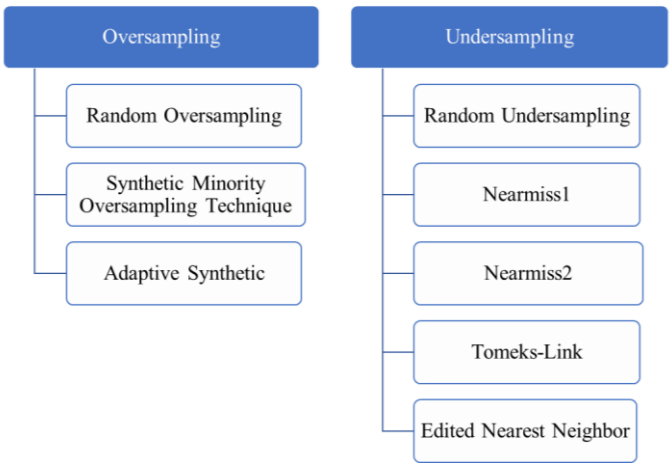

Figure 4.4. Classification of Sampling Approaches

Oversampling approaches

An oversampling is the method of procreating synthetic instance of minority class until the class size is same as majority instances. Some of the oversampling techniques are Adaptive Synthetic oversampling, Synthetic Minority Oversampling Technique and Random Oversampling approach. These oversampling techniques are discussed below in detail.

a) Adaptive Synthetic (ADASYN)

ADASYN also known as enhanced SMOTE, which modifies the minority instances concerning the weight of each class. The nearby class instance further generates the artificial instances of minority insider class (He et al., 2008).

b) Synthetic Minority Oversampling Technique (SMOTE)

SMOTE is a technique to produce the artificial synthetic data for handling the overfitting (Dittman et al., 2014). SMOTE applies the Regular Borderline to generate the malicious instance of minority class based on KNN and SVM (Bunkhumpornpat & Subpaiboonkit, 2013). The process of SMOTE in eqn. (1) shows that utilize instances of insider (x_i) to calculate the interpolation (x_{zi}) using KNN. It procreates the unique synthetic data proportional to minority instance (Gosain & Sardana, 2017).

$$x_{new} = x_i + \lambda(x_{zi} - x_i)_{zi} \quad (1)$$

Where,

x_{new} is a synthetically generated sample,

x_i is a minority instance of insider class,

x_{zi} is the interpolation of minority instance, and

λ is a constant value.

c) Random Over Sampling (ROS)

ROS is the process of procreating the instance of insider activities by reciting the minority class instance in random manner (Yap et al., 2014). As a result, machine learning may encounter overfitting.

Undersampling approaches

An undersampling is the method of eliminating the instance of majority class with respect to the class size of minority instances (Fujiwara et al., 2020). Random UnderSampling, Nearmiss1, Nearmiss2, Tomeks-Link, and Edited Nearest Neighbor are some of the well-known undersampling approaches. In below subsection, these approaches are discussed in detail.

a) Random UnderSampling (RUS)

RUS is the process of randomly minimizing majority instances of genuine user until the class size of minority instances of insider remains equals (Elhassan & Aljurf, 2016) (Hasanin & Khoshgoftaar, 2018). It results in losing critical information of the major genuine user instances. During classification, it led to misinterpretation.

b) Nearmiss (NM)

In Nearmiss, the process of undersampling is done by reducing the instance of genuine user with respect to other classes. Nearmiss has two different versions of sampling such as Nearmiss1 (NM-1) and Nearmiss2 (NM-2). In NM-1, the majority instances of a genuine class are considered only if the condition of nearby N minority instances satisfies the minimal in-between distance. In NM-2, the majority class of genuine instance are selected only if N's farthest instance of insider attain the least median.

c) Tomeks-Link (T-L)

The process of T-L is to lessen the majority instance and act as a classifier by eradicating the outlier (Elhassan & Aljurf, 2016). It witnesses the connection transpires based on the condition of closeness between two instances of different classes.

d) Edited Nearest Neighbor (ENN)

In ENN, the instances that fails to fit into the neighbor of KNN are diminished (Le & Zincir-Heywood, 2018). The analysis on working criteria of sampling approaches is depicted in table 4.9.

Table 4.9. Working Criteria of Sampling Approaches

S.no	Sampling technique	Working criteria
1	ROS	Random based
2.	SMOTE	KNN interpolation
3.	ADASYN	Adjacent class instance
4	RUS	Random based
5	NM-1	N adjacent insider threat instances
6	NM-2	N extreme insider threat instances
7	T-L	Link based
8	ENN	KNN based

The summary of oversampling and undersampling approaches are explained in table 4.10.

Table 4.10. Summary of Sampling Approaches

S.No	Sampling type	Definition	Sampling mechanism	Limitations	Techniques
1	Over sampling	The process of oversampling approaches is to generate the size of minority instance until the size remains same for majority class.	Prototype selection (controlled oversampling)	In case of increasing size of minority instance, the training time of ML model also increases.	Random Oversampling, SMOTE, and ADASYN
4	Undersampling	The process of reducing the instance of majority classes until the instance size equals the minority class.	Prototype selection (controlled or cleaning Undersampling)	Iteratively reducing the instance may result in encountering underfitting problem.	Random Under Sampling, Nearmiss1, Nearmiss2, Tomeks-Link, and Edited Nearest Neighbor.

Various undersampling and oversampling approaches for classification are explored to handle class imbalance problem in imbalanced data obtained from Data encoding. The performance metrics is used to choose the maximum performing sampling approach among various sampling approaches for classification.

In such case, existing study fails to identify the highly significant sampling approach, uprising a challenge in choosing the best sampling approach. The proposed methodology explored the eight sampling approaches to combat class imbalance problem. It is considered as one of the novelties of the proposed methodology. The sampling approach is evaluated using SVM classifier based on f-score, recall, precision, and accuracy.

Table 4.11 describes the performance metrics for evaluating the sampling approaches.

Table 4.11. Performance Metrics for Evaluating Sampling Approaches

Performance Metrics	**Description**	**Formula**
Precision	A fraction of samples exactly detected insider threats in the total number of insider threats. It is also known as maximum detection rate.	$Precision = \frac{TP}{(TP + FP)}$ (Or) $Precision = \frac{\text{Number of exactly detected insider activiti}}{(\text{Number of actual insider activities})}$
Recall	A fraction of insider samples exactly detected in the total number of predicted insider samples. It is also known as precise detection rate.	$Recall = \frac{TP}{(TP + FN)}$ (Or) $Recall = \frac{\text{Number of exactly detected insider activi}}{(\text{Number of predicted insider activities}}$
F-score	A defined harmonic mean between recall and precision.	$F - score = 2 \times \left(\frac{Precision \times Recall}{(Precision + Recall)}\right)$

Performance Metrics	Description	Formula
Accuracy	A percentage of exactly detected insider and genuine user samples. It is used to determine the overall performance of SVM.	$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)}$ (Or) $Precision = \frac{\text{Number of exactly detected activities}}{(\text{Number of all activities})}$

Where,

TP is True Positive,

TN is True Negative,

FP is False Positive, and

FN is False Negative.

The following are considered to achieve higher performing sampling approach. They are

- Maximum f-score,
- Minimal difference in recall and precision,
- Higher accuracy, and
- Precision is least important.

The sampled data is split into train and test data in the ratio of 80:20 to train SVM classifier. It was trained using train data and tested using test data. The simulation parameters for SVM is given in table 4.12.

Table 4.12. Simulation Parameters for SVM

Parameters	Values
Kernel	RBF
Gamma	0.001
Nu	0.02

The sampled data obtained from various sampling approaches mentioned earlier is used to train the SVM classifier and its performance is evaluated using four performance metrics. The best performance of SVM classifier implying eight sampling approaches is highlighted as the high-performing approach. Because, the best sampling approach effectively stabilize the instance of both genuine and insider. The performance of SVM using different sampling approaches based on four performance metrics is illustrated in Table 4.13.

Table 4.13. Results of Various Sampling Approaches.

Approaches Applied	**Accuracy (%)**	**F-score (%)**	**Precision (%)**	**Recall (%)**
Oversampling Approaches				
ADASYN	68.03	80	99	67
SMOTE	68.03	80	99	67
ROS	68.03	80	99	67
Undersampling Approaches				
ENN	68.03	80	99	67
NM-1	31.96	48	97	32
NM-2	**84.32**	**91**	**99**	**84**
RUS	71.66	80	99	71
T-L	68.03	80	99	67

From the table 4.13, the following are observed:

- For oversampling approaches, the performance remains same for ADASYN, SMOTE, and ROS with 68.03% accuracy, 80% f-score, 99% precision, 67% recall.
- Precisely detected genuine behavior samples are very minimal. Therefore, it is difficult to confront the imbalanced data using oversampling approaches.
- For undersampling approaches, NM-1 obtained only 48% f-score, and 31.96% accuracy of insider activity is inadequate.
- The outcome of T-L and ENN is similar with 68.03% accuracy, 80% f-score, 99% precision, and 67% recall is acceptable.

- NM-2 creates the synthetic insider threat instances with 84.32% accuracy, 91% f-score, 99% precision, and 84% recall, and outperformed ADASYN, ROS, and SMOTE. Consequently, it obtained the maximum outcome for all performance metrics mentioned earlier than oversampling approaches.

Sampled Data

In this section, NM-2 is selected for its highest performance among diverse sampling approaches based on the results obtained from table 4.13. The data representation before and after sampling using NM-2 is compared in Table 4.14.

Table 4.14. Before and After Undersampling Approaches

Training set	Before Sampling	After Sampling
Number of instances in genuine activity (Majority class)	(0, 39732)	(0, 268)
Number of instances in insider activity (Minority class)	(1, 268)	(1, 268)

Table 4.14 shows that 0 denotes instance of genuine activity, 1 denotes instance of insider activity. Before performing sampling, the number of genuine activity instances is 39732 and it is reduced to 268 after performing sampling. But the number of insider activity instance remains same before and after performing Data sampling. i.e., the instances of insider activities are reduced by the NM-2 using encoded data. It is evident that SVM classifier shows the highest performance using sampled data from NM-2 in the Data sampling based on higher f-score, precision and recall.

The limitations observed in NM-2 approach are listed below.

- Detecting insiders using limited logs (R3.1) is a major concern. R3.1 dataset contains the log information of the device, login, and HTTP, it is insufficient. Because, it satisfies only scenario-1 and scenario-2 of five CERT-driven insider threat scenarios for analyzing insider threat. Where scenario-1 is to utilize a USB drive after a working interval to acquire and manage sensitive documents in prohibited websites aiming for resignation. In scenario-2, sensitive data is obtained

from USB and visit business opponent websites for increasing recruitment possibilities in a condition of sensitive credentials

- The performance of NM-2 is higher than other sampling techniques by obtaining 84.32% accuracy. However, NM-2 fails to attain the best performance for insider threat detection. Hence, it is required to enhance the performance of Nearmiss2 by tuning the parameter. The limitations observed in Nearmiss2 is illustrated in figure 4.5.

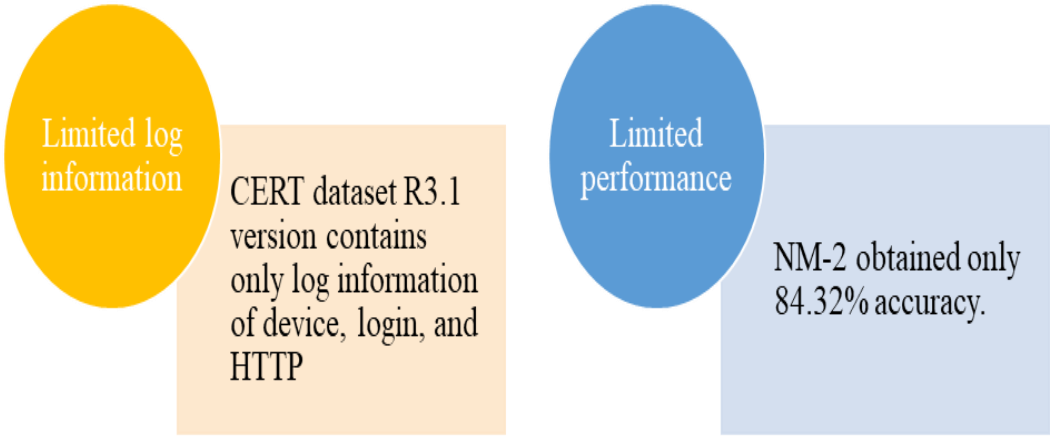

Figure 4.5. Limitations Observed in NM-2 Technique

It is analyzed that the limitations can be overcome by tuning NM-2 and evaluate the performance of NM-2 using diverse versions of CERT datasets.

Tuned Nearmiss2

Sampling using Nearmiss2 concentrates only on the outermost instance rather than the nearest one where the minority instances are resampled, that appears in the marginal outlier. Because, NM-2 samples the data based on two working conditions. They are

- Condition 1: minority class instance within N neighbors with the condition of satisfying minimal average distance between them. If the condition is satisfied, an instance of the majority class is selected.
- Condition 2: Meanwhile, minority class instances within N's outermost distance are considered to satisfy the minimal stipulated distance between them.

It only concentrates on outermost instance and are considered as one of the challenges. It can be handled by tuning the sampling strategy of Nearmiss2 which uses four sampling strategies to undersample the data. They are majority, not minority, not majority, and all. In 'majority', all the instances in the majority class would be resampled. In 'not minority', all the instances in every class except the minority class would be resampled. In 'not majority', all the instances in every class, excluding the majority class, are resampled. In 'all', every instance, irrespective of classes, are resampled. The Table 4.15 summarizes the NM-2 sampling strategy.

Table 4.15: Sampling Strategy of NM-2

S.No	Sampling strategy	Definition
1.	majority	Resample all the instances in the majority class.
2.	not minority	Resample all the instances in every class except the minority class.
3.	not majority	Resample all the instances in every class, excluding the majority class.
4.	all	Resample all the instances, irrespective of classes.

The above mentioned four sampling strategies would effectively balance the unequal distribution of data. In the proposed research, different versions of CERT datasets are used to evaluate the effectiveness of SVM with respect to four sampling strategies. The outcome of this tuned NM-2 is to effectively reduce the number of genuine activities equal to the number of malicious activities. The best performing sampling strategy is selected based on the performance of SVM classifier in terms of accuracy, precision, recall, and f-score. Table 4.16 shows the results of tuned Nearmiss2 using four versions of the CERT datasets.

Table 4.16. Results of Tuned Nearmiss2 using Four Versions of the CERT Datasets.

Dataset	Sampling strategy	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
R3.2	**majority**	**49.89**	**49.89**	**100**	**66.57**
	not minority	49.89	49.89	100	66.57
	not majority	99.72	0.0	0	0
	all	49.89	49.89	100	66.57
R5.1	**majority**	**74.39**	**100**	**48.78**	**65.57**
	not minority	73.78	100	47.56	64.46
	not majority	99.9	0	0	0
	all	72.86	100	45.73	62.76
R6.1	**majority**	**62.5**	**100**	**25**	**40**
	not minority	60.71	100	21.42	35.29
	not majority	99.88	0	0	0
	all	62.24	100	24.48	39.34
R6.2	**majority**	**62.24**	**100**	**24.48**	**39.34**
	not minority	60.71	100	21.42	35.29
	not majority	99.88	0	0	0
	all	61.98	100	23.97	38.68

From the table 4.16, the following are observed.

- The performance of Nearmiss2 with sampling strategy 'all' obtains zero precision, recall, and f-score for all four versions of the CERT dataset. Because it fails to recognize insider activity and not considered further.
- In R3.2, the performance of Nearmiss2 using three sampling strategies: ' majority', 'not minority', and 'all' is almost equal with 49.89% accuracy, 49.89% precision, 100% recall, and 66.57% f-score.
- However, in R5.1, the performance of Nearmiss2 with a 'majority' sampling strategy is higher in 74.39% accuracy, 100% precision, 48.78% recall, and 65.57% f-score, followed by 'not minority' and 'all'.

- Meanwhile, based on R6.1 and R6.2, Nearmiss2 obtained a little higher performance with a sampling strategy of 'majority' followed by 'all' and 'not minority.'

The above analysis shows that the performance of the sampling strategy 'majority' outperforms others irrespective of different versions of CERT datasets using the tuned Nearmiss2 undersampling approaches. It successfully handles the class imbalance problem. Table 4.17 shows results of Data sampling using tuned Nearmiss2.

Table 4.17. Results of Data sampling using Tuned Nearmiss2.

Dataset	Training set	Before sampling	After sampling
R3.2	Genuine behavior instance (Majority Class)	(0, 39732)	(0, 335)
	Insider behavior instance (Minority class)	(1, 335)	(1, 335)
R5.1	Genuine behavior instance (Majority class)	(0, 2,49,935)	(0, 234)
	Insider behavior instance (Minority class)	(1, 234)	(1, 234)
R6.1	Genuine behavior instance (Majority class)	(0, 2,50,078)	(0, 280)
	Insider behavior instance (Minority class)	(1, 280)	(1, 280)
R6.2	**Genuine behavior instance (Majority class)**	**(0, 2,50,078)**	**(0, 280)**
	Insider behavior instance (Minority class)	**(1, 280)**	**(1, 280)**

The following are observed from the table 4.17.

- The preprocessed data contains an equal number of genuine and malicious instances
- i.e., 335 per instance in R3.2, 234 per instance in R5.1, and 280 per instance in both R6.1 and r6.2 in a well-structured format that eases ML model training.
- It is understandable that the performance of SVM is enhanced by scrutinizing the sampling strategy to ‘majority’ in the Nearmiss2 sampling approach.

In the next section, the balanced data obtained from the tuned Nearmiss2 sampling approach is analyzed to identify insider threat patterns using the proposed hybrid machine

learning algorithm for detecting genuine, intentional and unintentional insiders. Dataset R6.2 is selected for further insider threat detection because it has updated log behaviors that satisfies all 5 Insider Threat scenarios. Next is classification, followed by Preprocessing.

Layer 2: Insider Detection

After Preprocessing, the classification is performed in this layer using balanced data obtained using tuned Nearmiss2 sampling approach. The following section discusses the proposed hybrid machine learning algorithm named B-SVM to detect and classify genuine, intentional and unintentional insider.

i) Proposed B-SVM for Classification

Balanced data obtained using tuned Nearmiss2 sampling approach that contains the log information of genuine and insider activities and is used for Insider Detection layer. The intentional insiders are individuals who deliberately perform malicious activities, while unintentional insiders carelessly expose the system to risks due to their negligence or mistakes. On the other hand, Genuine users exhibit behavior that is considered normal and non-threatening. Past research focuses only on detecting genuine and intentional insider, and fails to detect unintentional insider threats. Hence, a hybrid machine learning algorithm B-SVM is proposed which is capable of detecting all genuine, intentional and unintentional insiders with minimized misclassification rate.

The overview of B-SVM technique for detecting genuine, intentional and unintentional insiders is depicted in figure 4.6.

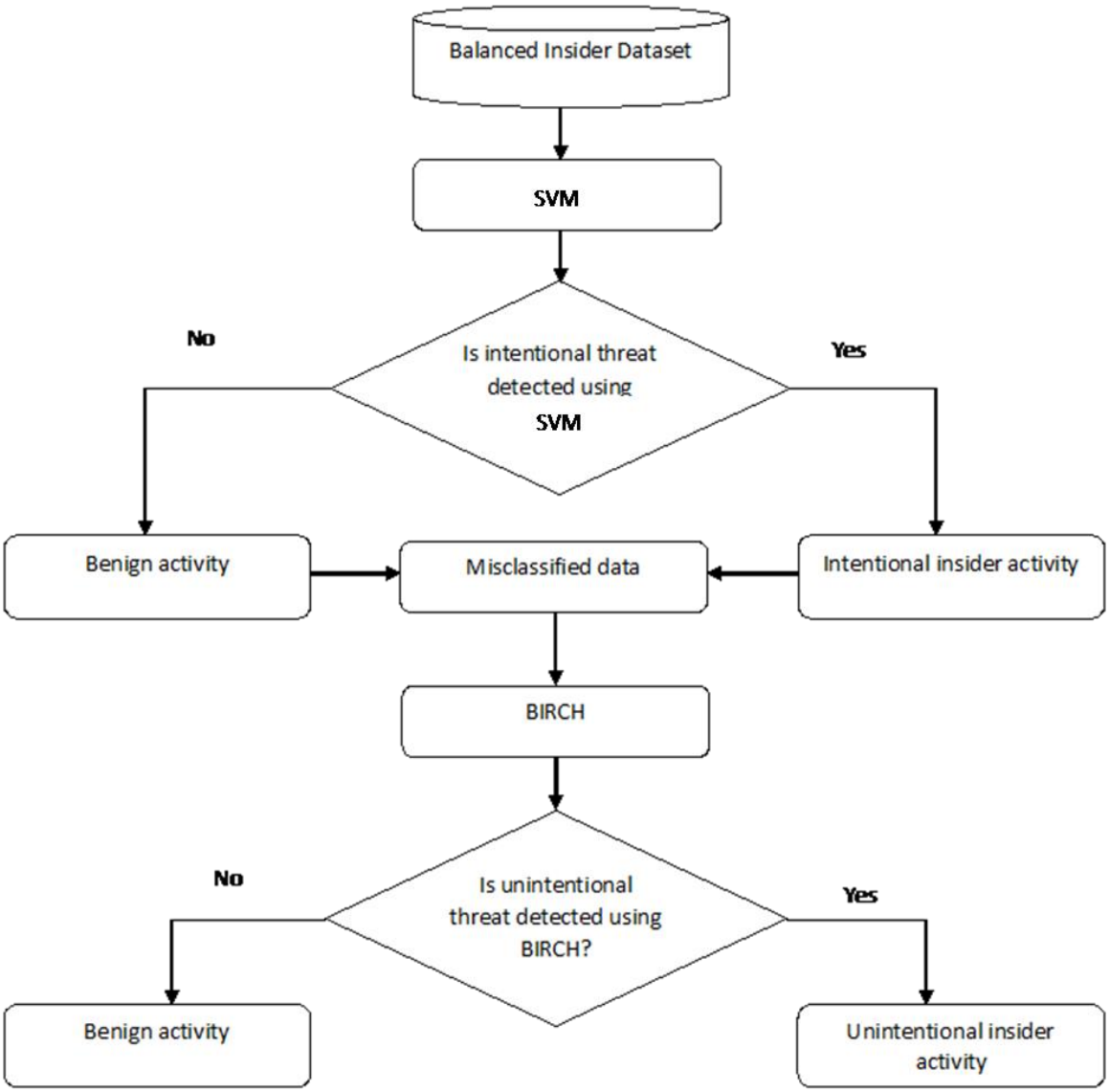

Figure 4.6. Overview of B-SVM Technique

Figure 4.6 shows that in B-SVM, an SVM model is first trained using the balanced data obtained from Preprocessing. SVM model detects and classify the user behavior into genuine and intentional insider activities. Where an abnormal behavior associated with malicious pattern is considered an intentional insider activity. The genuine and least unusual behavior rarely belonging to malicious patterns is considered genuine activity, thus increasing the false alarm in an SVM model. It results in misclassification and misinterpretation. Thus, the misinterpreted data from SVM model requires further anomaly detection since the false alarm rate is likely encountered.

The misclassified data from the SVM model is further trained using BIRCH model. Where it identifies the unusual malicious activity that diverges from normal genuine behavior based on similarity measures. It characterizes the sparse, abnormal user activity from normal behavior considered malicious.

The proposed B-SVM algorithm operates in a multi-stage process for detecting genuine, intentional and unintentional insiders which are discussed below.

1. ***Data Splitting****:* The initial step involves data splitting which is where the balanced data obtained from Preprocessing is split in the ratio of 80:20 for training and testing.
2. ***Classification Using SVM***: After Preprocessing, the Support Vector Machine (SVM) is deployed for classification. SVM model is a supervised learning algorithm that finds an optimal hyperplane that separates classes in a high-dimensional space. In this case, two functionalities can be achieved in this model.
    - The labeled dataset is used to train SVM, which differentiates intentional insider and genuine user. The SVM learns to classify data points based on different user behaviors during model training.
    - During prediction, the SVM model classifies new user activities into genuine user or potentially malicious intentional insider. This classification detects intentional insider threats based on predefined behavioral patterns.
3. ***Clustering using BIRCH****:*
    - **Handling Misclassified Data**: Although SVM is effective for classification, it can sometimes produce uncertain predictions, resulting in false positives (genuine users incorrectly classified as malicious) and false negatives (malicious users incorrectly classified as genuine). The B-SVM algorithm incorporates the BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) technique to handle these misclassified instances.
    - ***BIRCH Clustering:***
        - BIRCH creates a Clustering Feature (CF) Tree by grouping the data points based on their similarity. The BIRCH clustering process focuses

on the misclassified data from the SVM model, creating clusters that divulge patterns and characteristics of unintentional and genuine insider activities.

- The BIRCH algorithm examines the clusters created from the incorrectly classified data to find unique patterns of inadvertent insider behaviors. BIRCH identifies the situation where users might inadvertently engage in abnormal activity and differentiate them between typical user behavior variations and potential malicious actions. It detects two clusters they are (i) genuine users who exhibit unusual but non-malicious behavior, and (ii) unintentional insider who perform unintentional abnormal activities.

4. ***Decision-Making****:* Finally, B-SVM algorithm combines the result gained from BIRCH clustering and SVM classification. The hybrid decision-making is done by cross-referencing results from both methods to improve the model's reliability. By combining the strengths of SVM and BIRCH, the hybrid algorithm can more accurately classify users into genuine, intentional, and unintentional insiders. The analysis of decision using B-SVM is depicted in table 4.18.

Table 4.18: Analysis of Decision using Proposed B-SVM Algorithm

SVM	**BIRCH (B)**	**B-SVM**
Genuine	Genuine	Genuine
Genuine	Insider	Unintentional insider
Intentional insider	Genuine	Unintentional insider
Intentional insider	Insider	Intentional insider

Table 4.18 shows the following observations:

- If the user is considered as genuine by both SVM and BIRCH model, B-SVM classify such user as genuine.
- If the user is detected as insider by both SVM and BIRCH, then B-SVM classify the user as Intentional insider

- B-SVM model classify the user as unintentional insider in case user is detected as genuine by SVM or BIRCH.
- i.e., if the SVM model classifies an activity as potentially malicious but BIRCH clusters it with genuine user activities, this may indicate that the activity belongs to an unintentional insider rather than a deliberate threat.

The pseudo code of B-SVM technique for insider threat detection and classification is discussed below.

a) Pseudo code of B-SVM:

Input: User activity data (features: e.g., login time, file access, network traffic)

Step 1: Load the dataset

Step 2: The dataset is preprocessed

Step 3: An SVM classifier is trained using labeled data to detect genuine and intentional insiders.

The mathematical representation of SVM is explained below:

Linear SVM:

The hyperplane for a binary classification problem is expressed in eqn (2).

$$f(x) = w^T x + b \qquad (2)$$

Where:

- b is denoted as bias term,
- x is denoted as input feature vector, and
- w is denoted as weight vector.

The equation f(x)=0 is used to define the decision boundary.

Optimization Problem:

The SVM increases the boundary within two classes to find the optimal hyperplane is expressed in eqn (3). It is defined as the distance to the nearest data points (support vectors):

$$\min_{w,b} \frac{1}{2} \|w\|^2 \qquad (3)$$

Subject to Eqn (4):

$$y_i(w^T x_i + b) \geq 1, \forall i \qquad (4)$$

Where:

- $y_i \in \{-1,1\}$ are the class labels,
- x_i are the feature vectors.

Step 4: Use the trained SVM model to classify new user activity data into two categories:

- Genuine insiders (G)
- Intentional insiders (I)

Step 5: Extract falsely detected intentional insider activities

Step 6: Preprocess falsely_detected_intentional, where normalization and transformation are required

Step 7: Build a BIRCH clustering model using falsely detected intentional insider activities to build a BIRCH clustering model.

The mathematical representation of BIRCH is explained below.

BIRCH

- *Clustering Feature (CF):*

In CF, each node contains clustering feature which summarizes a group of data points in a cluster. The CF for data points in a cluster is defined as (N,LS,SS).

Where,

- N denotes a total number,
- LS denotes a linear sum is expressed in eqn (5).

$$LS = \sum_{i=1}^{N} x_i \qquad (5)$$

- SS denotes squared sum is expressed in eqn (6).

$$SS = \sum_{i=1}^{N} {x_i}^2 \qquad (6)$$

Using CF, the centroid μ and radius R for each cluster is calculated as:

- Centroid is expressed in Eqn (7).

$$\mu = \frac{LS}{N} \qquad (7)$$

- Radius is expressed in Eqn (8)

$$R = \sqrt{\frac{SS}{N} - \|\mu\|^2} \qquad (8)$$

- *Distance Metrics:*

BIRCH uses distance metrics to decide whether to merge a new data point into an existing cluster or create a new cluster. Common distance metrics includes Centroid Distance, Average Inter-cluster Distance, and Maximum Diameter.

- *CF Tree Construction:*

The CF Tree is constructed incrementally by:

1. Inserting new data points into the closest leaf node.
2. Updating the CF of the leaf node.
3. Splitting the leaf node if it exceeds the predefined threshold (branching factor BBB and threshold T).
4. Threshold Condition: The threshold T determines the maximum allowable diameter of a cluster: $R \leq T$

   Where Clusters that violate this condition trigger node splitting.

5. Final Clustering (Global Clustering):

   After constructing the CF Tree, a global clustering method (e.g., hierarchical or k-means) in leaf entries is employed to produce the ending clusters. BIRCH

clusters data by summarizing it using compact mathematical structures (CFs), maintaining computational efficiency and scalability, especially for large datasets.

Step 8: Use the BIRCH model to identify and label clusters of similar activities as Genuine insiders (G), Intentional insiders (I), Unintentional insiders (U)

Step 9: Use the labeled clusters to classify new user activity data into three categories:

- Genuine insiders (G)
- Intentional insiders (I)
- Unintentional insiders (U)

Output: final classification as Genuine (G), Intentional (I), or Unintentional (U)

4.3 EXPERIMENTAL RESULTS

4.3.1 PERFORMANCE METRICS FOR B-SVM

The balanced data obtained from Preprocessing is analyzed using B-SVM technique for insider threat detection. The performance of B-SVM is evaluated based on nine performance metrics for detecting genuine, intentional and unintentional insiders. The metrics include accuracy, precision, f-score, recall, misclassification rate, True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR), False Negative Rate (FNR) based on true positive (TP), true negative (TN), false positive (FP), false negative (FN) from confusion matrix.

FPR is a percentage of incorrectly recognized genuine activities. Meanwhile, FNR is a percentage of incorrectly identified intentional malicious activities. The percentage of TNR should remain higher than FPR and FNR. i.e., a percentage of precisely detected intentional malicious activities. Table 4.19 illustrates significant performance metrics for evaluating the performance of B-SVM for insider threat detection.

Table 4.19. Performance Metrics to Evaluate B-SVM for Insider Threat Detection.

Performance Metrics	**Description**	**Formula**

Precision	Rate of precisely detected malicious samples considered as malicious insiders from entire insider threat data.	$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$
Recall	The rate of exactly identified genuine samples also called genuine users from entire samples.	$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$
F-score	A harmonic mean between precision and recall.	$\text{F} - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
Accuracy	Utilized incorrectly recognized genuine and malicious samples to calculate the harmonic mean between precision and recall.	$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$
Misclassification on rate	Rate of imprecisely recognized both genuine and malicious activities.	$\text{Misclassification rate} = \frac{FP + FN}{TP + TN + FP + FN}$
FPR	Rate of incorrectly detected genuine activities.	$\text{FPR} = \frac{FP}{TN + FP}$
FNR	Rate of incorrectly recognized malicious activities.	$\text{FNR} = \frac{FN}{TP + FN}$
TNR	Rate of correctly detected genuine activities	$\text{TNR} = \frac{TN}{TN + FP}$
TPR	Rate of correctly recognized malicious activities	$\text{TPR} = \frac{TP}{TP + FN}$

Accuracy, f-score, and misclassification rate are highly utilized in B-SVM model evaluation compared to precision and recall. Past studies fail to consider misclassification rate for intentional and unintentional insider threat detection. The present study applies all metrics mentioned above and prefers the misclassification rate (or false alarm rate) for a significant evaluation.

4.3.2 ELABORATIVE RESULT ANALYSIS OF THE P&ID PHASE

The performance of P&ID phase is evaluated using CERT insider dataset and the CIC darknet datasets. The following subsection discusses the experimental results of P&ID phase containing Preprocessing in layer 1 and Insider Detection in layer 2.

Layer 1: Preprocessing

The performance of Preprocessing is evaluated using CERT insider and CIC darknet datasets. The following subsection explains the result obtained from three stages of Preprocessing techniques namely Data integration, Data encoding and Data sampling using both datasets.

i) Data integration

The following section discusses the result obtained in Data integration using the CERT insider dataset and CIC darknet dataset.

a) CERT Insider Dataset

Table 4.20 shows the result of the Data integration using the simple feature concatenation technique combines log information gathered from five sources namely device.csv, http.csv, network.csv, email.csv, and logon.csv.

Table 4.20. Result of Data Integration using Simple Feature Concatenation Technique in CERT Insider Dataset.

Feature name	**Feature Details**
insider threat	It defines user activity in terms of genuine or malicious
source	It defines the source where activity information is obtained
date	It defines the date performing the particular event
user id	It defines the unique id of user who performs the particular activity
pc id	It defines the unique id of computer where particular activity is accomplished
activity	It defines the activity being performed

The above table outlines the features of integrated data after performing Data integration using CERT insider dataset. After integration, the integrated data contains four standard features which is common in diverse sources. i.e., date, user id, pc id, and activity. In addition, two features such as 'source' and 'insider threat' that specifies origin's source and activity class is present in the integrated data.

b) CIC Darknet Dataset

Data integration for CIC darknet dataset is not necessary. Because, the dataset is already fully consolidated with 85 relevant features. This lack of integration minimizes the Preprocessing techniques and focus on further Data encoding and sampling since the dataset contains categorical values and imbalanced data.

ii) Data encoding

After performing Data integration, Data encoding is incorporated using integrated data. The result of Data encoding using the CERT insider dataset and CIC darknet dataset is explained below.

a) CERT insider dataset

The results of Data encoding using categorical encoding in the CERT insider dataset are mentioned in Table 4.21.

Table 4.21. Result of Data Encoding using Categorical Encoding Technique in CERT Insider Dataset.

Attribute(s)	**Before encoding**		**After encoding**	
	Datatype	**Value**	**Datatype**	**Value**
insider threat	Categorical	Genuine, malicious	Integer	0,1
source	Categorical	Logon, device, http	Integer	0,1,2
date	Ordinal	07-01-2010 02:23:00	Integer	1280707200
user id	Categorical	CCH0959	Integer	4
pc id	Categorical	PC-0588	Integer	128
activity	Categorical	http://linkedin.com/jobs/displayhome.html	Integer	750

Table 4.21 shows that every data relating to a user, PC, and activity is converted into a numerical representation. Meanwhile, the date is converted into several milliseconds, providing encoded data from a standard record.

b) CIC Darknet dataset

The results of Data encoding based on categorical encoding using the CIC darknet dataset are mentioned in Table 4.22.

Table 4.22. Result of Data Encoding using Categorical Encoding Technique in CIC Darknet Dataset.

Attribute(s)	**Before encoding**		**After encoding**	
	Datatype	**Value**	**Datatype**	**Value**
Flow ID	Categorical	10.135.135.11-216.58.220.99-57158-443-6	Integer	20738
Src IP	Categorical	10.135.135.11	Integer	13
Dst IP	Categorical	216.58.220.99	Integer	2973
Date	Ordinal	24/07/2015 04:09:48 PM	Integer	4532
Label	Categorical	Non-Tor, NonVPN, VPN, Tor	Integer	0, 1, 2, 3
Label.1	Categorical	P2P, Browsing, Audio-Streaming, Chat, File-Transfer, Video-Streaming, Email, VOIP	Integer	0, 1, 2, 3, 4, 5, 6, 7

From Table 4.22, it is observed that several categorical features from the CIC darknet dataset, namely ‘Flow ID’, ‘Src IP’, ‘Dst IP’, ‘Label, and ‘Label.1’, were transformed into numerical values through categorical encoding techniques. Before encoding, ‘Date’ is ordinal data and converted into number of milliseconds.

iii) Data sampling

After Data encoding, sampling is performed using the tuned Nearmiss2 sampling to handle the class imbalance problem. The results obtained from Data sampling using both the CERT insider dataset and the CIC darknet dataset is discussed below.

a) CERT insider dataset

The CERT insider dataset has a target variable, "insider threat", for binary classification. The genuine instances are the majority class, with 250,078 instances, while malicious instances are the minority class, with 280 instances. Table 4.23 shows the result of tuned Nearmiss2 based sampling using CERT insider dataset.

Table 4.23: Result of Data Sampling using Tuned Nearmiss2 Approach in CERT Insider Dataset.

Training Set	**Before Sampling**	**After Sampling**
Genuine behavior Majority instance	(0, 2,50,078)	(0, 280)
Malicious behavior minority instance	(1, 280)	(1, 280)

Table 4.23 shows that Nearmiss2 sampling has reduced the 2,50,078 genuine instances to 280 instances. The instance of malicious behavior remains same before and after sampling.

b) CIC darknet dataset

The CIC Darknet dataset's target variable "Label" includes four classes: Non-Tor, NonVPN, VPN, and Tor. Within these, Non-Tor and NonVPN were majority classes, while VPN and Tor were minority classes. There are 110,442 records labeled as Non-Tor, 23863 records as NonVPN, 22919 records as VPN and 1392 records as Tor. Table 4.24 shows the result of tuned Nearmiss2 based sampling using the CIC darknet dataset.

Table 4.24: Result of Data Sampling using Tuned Nearmiss2 Approach in CIC Darkent Dataset.

Training set	Before sampling	After sampling
Genuine behavior instance (Majority Class, Non-Tor instances)	(0, 110442)	(0, 1392)
Genuine behavior instance (Minority Class, NonVPN instances)	(1, 23863)	(1, 23863)
Malicious behavior instance (Majority class, VPN instances)	(2, 22919)	(2, 1392)
Malicious behavior instance (Minority class, Tor instances)	(3, 1392)	(3, 22919)

From the table 4.24, the following are observed:

- After sampling, the majority class Non-Tor instances are reduced to 1392 instances.
- Minority class NonVPN contains 23863 instances which is same after sampling.
- Majority class VPN instances are reduced to 1392 instances after sampling.
- Minority class Tor instances are increased to 22919 instances after sampling.

Layer 2: Insider Detection

After Data sampling, the Insider Detection is performed using balanced data obtained from Preprocessing layer. B-SVM technique is used to classify and detect genuine, intentional and unintentional insiders. The following subsection discusses the result obtained from B-SVM for Insider Detection using both CERT insider and CIC darknet datasets.

i) B-SVM for classification

The following Table 4.25 gives the performance evaluation of proposed B-SVM in the P&ID phase with SVM and BIRCH using CERT insider and CIC darknet datasets. It is

evaluated based on precision, recall, f-score, accuracy, misclassification rate, FPR, FNR, TNR, and TPR metrics.

Table 4.25: Result of Proposed B-SVM in the P&ID Phase

Dataset	Algorithm	Precision (%)	Recall (%)	F-score (%)	Accuracy (%)	Misclassification rate (%)	FPR (%)	FNR (%)	TNR (%)	TPR (%)
CERT insider dataset	SVM	88	96.03	91.84	91.46	8.53	12	4.36	88	95.63
	BIRCH	100	33.34	50	66.67	33.33	0	40	100	60
	B-SVM	**100**	**98.01**	**98.99**	**99.01**	**0.99**	**0**	**1.94**	**100**	**98.05**
CIC darknet dataset	SVM	95.7	95.42	95.39	95.42	4.58	1.51	4.79	98.48	95.2
	BIRCH	67.15	54.25	49.33	54. 25	45.75	15.24	45.74	84.75	54.25
	B-SVM	**95.90**	**95.69**	**95.59**	**95.69**	**4.31**	**1.44**	**4.52**	**98.55**	**95.47**

From the above table 4.25, the following observations are noted:

- SVM on the CERT dataset showed a relatively high misclassification rate of 8.53%, with 12% FPR and 4.36% FNR.
- On the CERT dataset, BIRCH confined with recall and F-score and obtained a high misclassification rate (33.33%).
- B-SVM on the CERT dataset outperformed both SVM and BIRCH with the highest precision (100%), recall (98.01%), F-score (98.99%), and accuracy (99.01%) with minimal misclassification rate (0.99%).
- On the CIC darknet dataset, the performance of B-SVM surpassed SVM and BIRCH by obtaining the highest precision (95.90%), recall (95.69%), f-score (95.59%), and accuracy (95.69) with lowest misclassification rate (4.31%).

The following table 4.26 gives the mean performance of proposed B-SVM in the P&ID phase for Insider Detection. It is evaluated based on precision, recall, f-score, accuracy, misclassification rate, FPR, FNR, TNR, and TPR.

Table 4.26: Mean Performance of Proposed B-SVM in the P&ID Phase

Dataset	ML algorithms	Precision (%)	Recall (%)	F-score (%)	Accuracy (%)	Misclassification rate (%)	FPR (%)	FNR (%)	TNR (%)	TPR (%)
CERT insider	B-SVM	100	98.01	98.99	99	0.99	0	1.94	100	98.05
CIC darknet		99.32	99.31	99.3	99.31	0.69	0.28	0.28	99.71	96.39
Mean Performance		**99.66**	**98.66**	**99.14**	**99.15**	**0.84**	**0.14**	**1.11**	**99.85**	**97.22**

From the above table 4.26, the following observations were noted:

1. The proposed B-SVM a mean precision of 99.66%, a mean recall of 98.66%, and a mean F-score of 99.14%, with an overall accuracy of 99.15%.

2. The misclassification rate was negligible at 0.84%, while FPR and FNR were 0.14% and 1.11%, respectively.

3. The high true negative rate (TNR) of 99.85% and true positive rate (TPR) of 97.22% help further validate B-SVM's reliability in binary and multi-class classification.

4.3.3 COMPARISON OF PROPOSED B-SVM WITH EXISTING METHODS

Table 4.27 compares the average performance of the P&ID phase with existing techniques after applying B-SVM. The performance of the P&ID phase is evaluated using CERT and CIC Darknet datasets based on precision, recall, f-score, accuracy, and misclassification rate.

Table 4.27: Performance Evaluation of B-SVM in the P&ID Phase vs Other State-of-art-methods

Study	Method	Dataset	Results				
			Precision (%)	Recall (%)	F-score (%)	Accuracy (%)	Misclassification rate (%)
Proposed P&ID	**B-SVM**	**CERT & CIC Darknet**	**99.66**	**98.66**	**99.14**	**99.15**	**0.85**
Prasad et al. (2024)	ENN+ AdaBoost + PCA+ Ensemble Learning	CERT	-	-	-	99	-
Singh et al. (2023)	SVM+GA	CERT	85.7	84.4	84.85	88	-
Asha et al. (2022)	OCSVM	CERT	64.92	100	78.72	82.46	17.54
Al-Mhiqani et al. (2022)	Random Forest	CERT	51	50	49	90	10
Garba et al. (2021)	Kmeans+ PCA	CERT	89	-	-	-	-
Al-Shehari et al. (2021)	CGAN	CERT	79.12	76.07	74.85	-	-
Mohammed et al. (2021)	LightGBM	CERT	-	-	70.42	98.03	1.97
Noever (2019)	Random Forest	CERT	-	-	-	98	2
Ferreira et al. (2019)	Random Forest	CERT	96.52	-	75.42	-	-
F.Janjua et al. (2018)	Adaboost	TWOS	-	98	-	98.3	1.7
Le & Zincir. (2018)	SOM	CERT	71.77	-	-	88.38	11.62
Chattopadhyay et al. (2018)	Random Forest	CERT	65.6	80.71	70.61	-	-

From the table 4.27, it is observed that:

- The proposed method achieves high precision and recall of 99.66% and 98.66%, ensuring accurate detection of almost all genuine instances and malicious behaviors and outperforms other existing methods.
- The proposed method achieves the highest f-score (99.14%) and accuracy (99.15%) among existing methods.
- The misclassification rate of B-SVM (0.99%) is significantly lower than other models.

From the above findings, it is evident that the proposed method surpasses the existing methods for insider threat detection.

4.3.4 INSIDER DETECTION USING B-SVM

The following figure 4.7 depicts the outcome of P&ID phase after applying proposed B-SVM to detect genuine, intentional and unintentional insiders on CERT insider and CIC darknet datasets.

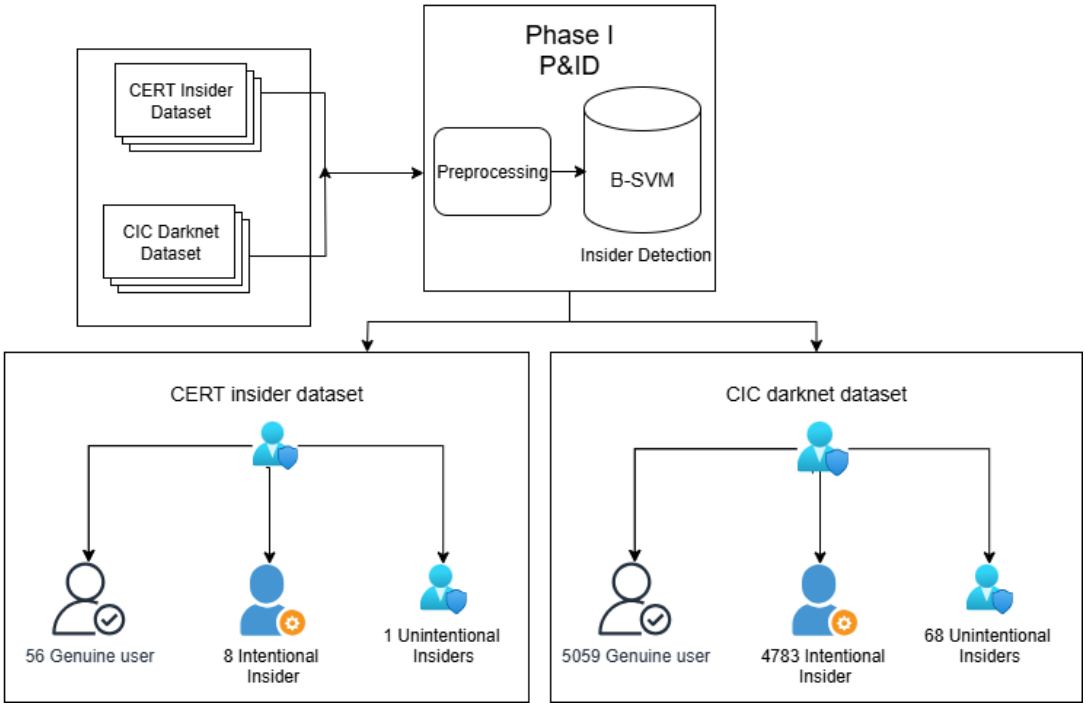

Figure 4.7: Results of P&ID Phase After Applying B-SVM using CERT Insider and CIC Darknet Datasets

The insiders are detected in P&ID phase using proposed B-SVM. It is observed that P&ID phase classifies the user into genuine, intentional and unintentional insider. B-SVM detected 56 genuine users, eight intentional insiders and one unintentional insider with CERT insider dataset. In CIC darknet dataset, B-SVM detected 5059 genuine users, 4783 intentional insiders and 68 unintentional insiders using P&ID phase.

4.4 OUTCOME OF PHASE I

The outcome of the P&ID phase is listed below.

- Extensive preprocessing such as integrating log detail and encoding into numerical data, has been performed.
- In Nearmiss2, the Majority sampling strategy ensured a balanced dataset and tuned the accuracy of the machine learning model for insider threat detection.
- The proposed B-SVM attained higher accuracy in detecting insider threats. Using the CERT dataset, the proposed B-SVM identified eight intentional insider activities and one unintentional insider activity among 250,078 daily log entries.
- On the Darknet dataset, the proposed B-SVM phase detected the activities of 4,783 intentional-Darknet users and 68 unintentional-Darknet users such as VPN (42 users), Tor (21 users) and NonVPN (5 users) among 134,305 daily activities.
- Intentional and unintentional insiders are successfully detected and classified using the proposed B-SVM algorithm with 99.15% accuracy and minimal misclassification rate. Thus achieving the stated secondary objective 1.

4.5 LIMITATION OF PHASE I

Phase I has the Following limitation.

- Since, the B-SVM model is limited to detecting intentional and unintentional insiders, this phase does not offer a mechanism for mitigating detected insider threats. So, a mechanism is required for mitigating both unintentional and intentional insider threats.

4.6 CHAPTER SUMMARY

Initial Preprocessing, including Data integration and Data encoding, is done. Then, the sampling techniques are explored and incorporated for the class imbalance problem taken for study, such as Random OverSampling, SMOTE, ADASYN, Random UnderSampling, Edited Nearest Neighbor, Nearmiss1, Nearmiss2 and Tomeks' link, are discussed. Next to sampling is classification. For classification, a proposed B-SVM model is proposed for detecting genuine, intentional and unintentional insiders. This chapter detected and classified intentional and unintentional insiders. However, detection without mitigation increases the chance of risk in security systems. Thus, there is a significant need to mitigate both intentional and unintentional insiders which are elaborately discussed in chapter 5 and 6. The chapter 5 discusses the mitigation of unintentional insider elaborately.